

```

---
title: "Todd_Garner_DS6306_Week6_FLS"
author: "Todd Garner"
date: "2023-02-05"
output:
  beamer_presentation: default
  powerpoint_presentation: default
---

```

```

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = FALSE)
```

```

For Live Session Week 6 Part 1 - **Download the training set: Connect to the opendatasoft website and download the random sample of 891 Titanic Passengers. This is the training set. The data come in JSON form format and you can use this URL to access the data:**

<https://public.opendatasoft.com/api/records/1.0/search/?dataset=titanic-passengers&rows=2000&facet=survived&facet=pclass&facet=sex&facet=age&facet=embarked>

Hint: This is not trivial. I recommend that you use the jsonlite package (fromJSON()) and RCurl package (getURL()) to access the data. (We covered this in Unit 4).

1. *Try your best to access the data using the URL. You may also find the data (titanic_train.csv) on github. We will go over this data ingestion in live session.*

```

```{r}
library(jsonlite)
library(RCurl)

url <- read.csv(file.choose(), header = TRUE)
head(url)
summary(url)
str(url)
sum(url$SibSp)
sum(url$Parch)
```

```

When I simply copy and paste the URL into a browser, here is the response from the site:

```

"{
  "error": "Unknown dataset: titanic-passengers"
}"

```

Not a good sign. I wonder if the link is valid? I'll try shorter elements of the URL to see if I can navigate to the location of the data set. I navigated to public.opendatasoft.com and attempted numerous combinations to find the data set to no avail. I'm going to load the .csv file. I notice that it is unclear which column shows survived or died. SibSP and Parch are binary, 1s and 0s. There are 187 SibSP values and 164 Parch values - those with 1s. From a brief search, there were a total of 2,208 souls aboard and 712 survivors. As there are 418 observations in our data set, a subset, this doesn't shed any light on that question. If you total those two, it equals 351. Again, a subset of 418.

I was able to search for and found the data set on www.kaggle.com. It had the data set broken down by columns. Of which, the Survived column has been deleted from the "test" data set. Hmm....I think I will download the data set from Kaggle because otherwise, answering #2 will prove challenging. Here are the column names in the kaggle training data set:

```

PassengerId: unique ID of the passenger
**Survived: 0 = No, 1 = Yes**
Pclass: passenger class 1 = 1st, 2 = 2nd, 3 = 3rd
Name: name of the passenger
Sex: passenger's sex
Age: passenger's age
SibSp: number of siblings or spouses on the ship

```

Parch: number of parents or children on the ship
Ticket: Ticket ID
Fare: the amount paid for the ticket
Cabin: cabin number
Embarked: Port of embarkation (C = Cherbourg, Q = Queenstown, S = Southampton)

It's only now that I notice there are two titanic data sets in the GitHub repo. Details apparently do matter. So, I will work with these two data sets.

Here is some code I gathered from ChatGPT by asking the following question: ****In the widely used data set titanic-passengers, in R there are two sets. a training set with 891 rows and 12 columns that includes a column of Survived. It has a 1 or a 0. 1 = survived, 0 = died. The question is first, how to change the 1 and 0 to "Lived" or "Died" and then to use K-NN to classify those who survived and died based on "Age" and "Class." Class has 3 choices, 1 = 1st class, 2 = 2nd, 3 = 3rd. please right this code in R****

I'm posting the code generated by ChatGPT in markdown language below:

```
# Load the Titanic dataset
data("Titanic")

# Change the values of the "Survived" column to "Lived" or "Died"
Titanic$Survived = ifelse(Titanic$Survived == 1, "Lived", "Died")

# Create a new dataset with only the "Age" and "Class" columns
Titanic_subset = Titanic[, c("Age", "Class")]

# Impute missing values in the "Age" column with the median
Titanic_subset$Age = ifelse(is.na(Titanic_subset$Age),
                           median(Titanic_subset$Age, na.rm = TRUE), Titanic_subset$Age)

# Split the data into a training set and a test set
set.seed(123)
train_index = sample(1:nrow(Titanic_subset), floor(0.8 * nrow(Titanic_subset)))
train = Titanic_subset[train_index, ]
test = Titanic_subset[-train_index, ]

# Fit a K-NN model on the training set
library(class)
knn_model = knn(train[, -2], test[, -2], train[, 2], k = 3)

# Predict the survival of passengers in the test set
pred = knn_model

# Evaluate the accuracy of the model
mean(pred == test[, 2])
```

I will have to modify the code to load the training data set.

Many hours later after trying to mold the ChatGPT code into a workable solution, discarding it and following an article on Edureka.co, I've thrown up my hands. The Edureka explanation makes perfect sense but I'm just not able to get it to work. So, I'm back here at the start to contemplate, "What is it that I'm really trying to accomplish?" I've used the Kaggle data sets and I've defaulted back to the .csv files in the GitHub repo. So, I'm on #2.

****Use KNN to classify those who survived and died based on Age and class.**** In other words, find a way to model whether a passenger lived or died based on Age and Pclass (1st, 2nd or 3rd class passage). Solving for lived or died based on these two columns and the values within them. I will search my code to see where I've gone wrong.

```
```{r}
Load the Titanic dataset
Titanic <- read.csv(file.choose(), header = TRUE)
head(Titanic)

#data("Titanic")
```

```
#Scaling is something that makes sense to me as there are three distinct Pclasses but a wide
range of Ages...a continuous variable. Should I scale the Pclass values as well? It would
put them all in the same basic range of values and bring the Euclidian distances more in
line. So, I'll give it a try.
#Scaling Age so it is more in line with the passenger class column.
#Titanic[,6] <- scale(Titanic[,6])
#head(Titanic)
#Titanic[, 3] <- scale(Titanic[, 3])
#head(Titanic)
```

#Okay, now that the scaling is done on those two columns, I will create a new data.frame with only those two columns. But, is that correct? What about the Survived column? It needs to be in there as well. This is something I didn't do on my first time through.

#After reading through the article on Edureka.co ( <https://www.edureka.co/blog/knn-algorithm-in-r/>), i see that they've removed the dependent variable so I'll remove "Survived" from the training set.

```
Create a new dataset with only the "Age", "Survived" and "Class" columns
library(dplyr)
Titanic_new <- Titanic %>% select(Survived, Pclass, Age)
head(Titanic_new)
nrow(Titanic_new)
Titanic_exp <- Titanic %>% select(Pclass, Age)
head(Titanic_exp)
nrow(Titanic_exp)
```

# I now see that I have some NAs in the Age column. That explains why it was throwing errors in my KNN function. "No missing valued allowed." I need to delete the rows, of all columns based on the NA's in the Age column.

```
Titanic_new_final <- Titanic_new[complete.cases(Titanic_new),]
summary(Titanic_new_final)
nrow(Titanic_new_final)
```

```
Titanic_final <- Titanic_exp[complete.cases(Titanic_exp),]
summary(Titanic_final)
nrow(Titanic_final)
```

...

### Holy smokes! That removed 177 rows! That explains volumes of why I was having issues before. Well, better late than never. Let's roll on.

Previously, I had loaded up ChatGPT with the question. And, I likely posed the wrong question which is why I just flailed. The comments below were from hours ago. I'll leave them there for reference.

I had to modify quite a bit to make it work. Part of the problem is that I worded the question incorrectly and it assumed I would use the training set in two subsets. I will have to repair this error. I could resubmit my question correctly, but I think I can work with this. On further reflection, perhaps it's not such a bad idea to use the training set, broken down 80/20 to train the model. I can then accurately test the validity of the model. Then, I'll run the full test set through and see what the results are.

UPDATE: I'm moving on with my code below.

```
` `{r}
```

#I'm using the titanic\_train.csv as my training AND testing set for now. I will insert the real titanic\_test.csv once the model is working.

# Split the data into a training set and a test set. Setting the training set at 80% and the test set at 20% of the train\_set.csv. The data set that includes the column Survived.

```

set.seed(123)
train_new_index = sample(1:nrow(Titanic_new_final), size = nrow(Titanic_new_final)*0.8,
replace = FALSE)
train_new = Titanic_new_final[train_new_index,]
test_new = Titanic_new_final[-train_new_index,]
View(train_new)

set.seed(123)
train_index = sample(1:nrow(Titanic_final), size = nrow(Titanic_final)*0.8, replace = FALSE)
train = Titanic_final[train_index,]
test = Titanic_final[-train_index,]

str(train)
str(test)
head(train)
head(test)

#There are 571 observations (rows) in the "training" set and 143 in the "test" set. The
Edureka.co suggestion was to use a k value that is the square root of the number of
observations in the training set. That's close to 24. I'll try 23 and 24 to see what
happens.

Fit a K-NN model on the training set
#Creating separate dataframe for 'Creditability' feature which is our target.
train.life_labels <- data.frame(Titanic_final[train_index,1])
test.life_labels <- data.frame(Titanic_final[-train_index,1])

#View(train.life_labels)
library(class)
dim(train)
dim(test)
dim(train.life_labels)
dim(test.life_labels)
head(train.life_labels)
head(test.life_labels)
View(train)
View(test)
View(train.life_labels)
nrow(test)
nrow(train)
nrow(train.life_labels)

This is looking promising. Amazing what thinking can do!
library(caret)

knn.23 <- knn(train, test, train_new$Survived, k = 23)
knn.23
knn.24 <- knn(train, test, train_new$Survived, k = 24)
knn.24

table(test_new$Survived, knn.23)
confusionMatrix(table(test_new$Survived, knn.23))

table(test_new$Survived, knn.24)
confusionMatrix(table(test_new$Survived, knn.24))
#knn_23 <- knn(train, test,

#I continue to experience the following error:

> knn.23 <- knn(train, test, train.life_labels, k = 23)
#Error in knn(train, test, train.life_labels, k = 23) :
'train' and 'class' have different lengths

```

#But, when I run the dim() and nrow() function, here is what I get:

```
#> dim(train)
#[1] 571 2
#> dim(test)
#[1] 143 2
#> dim(train.life_labels)
#[1] 571 1
#> dim(test.life_labels)
#[1] 143 1
#> nrow(test)
#[1] 143
#> nrow(train)
#[1] 571
#> nrow(train.life_labels)
#[1] 571
```

#The dimensions are different only in the number of columns, 2 versus 1. But, it says, "train" and "Class" have different lengths, not different dimensions. Is this truly causing the breakage?

...

Sweet Mama! I did it!!!! Man, that was laborious! But, it feels good to have solved it.

Okay, now onto #3.

## \*\*Use your age and predict your survival based on each of the ticket classes.\*\*

```
`{r}
```

#This is the same code as run above

```
knn.23 <- knn(train, test, train_new$Survived, k = 23)
knn.23
knn.24 <- knn(train, test, train_new$Survived, k = 24)
knn.24
```

```
table(test_new$Survived, knn.23)
confusionMatrix(table(test_new$Survived, knn.23))
```

```
table(test_new$Survived, knn.24)
confusionMatrix(table(test_new$Survived, knn.24))
```

#At this point, I'm attempting to put just a single data point through the model to test whether based on my Age and Pclass, if I would survive or perish. I backed up and ran my model with it unscaled and it still works but the accuracy is down a bit. Makes sense. Now, I'm running the predict() function through and trying to pass it my Age and it's saying it's a factor.

```
test_new <- data.frame(Pclass = 1, Age = 59)
test_new
```

```
test_new.2 <- data.frame(Pclass = 2, Age = 59)
test_new.2
```

```
test_new.3 <- data.frame(Pclass = 3, Age = 59)
test_new.3
```

```
knn.todd_1 <- knn(train, test_new, train_new$Survived, k = 24)
knn.todd_1
```

```
knn.todd_2 <- knn(train, test_new.2, train_new$Survived, k = 24)
knn.todd_2
```

```
knn.todd_3 <- knn(train, test_new.3, train_new$Survived, k = 24)
knn.todd_3
```

```
```
```

It's looking bad for me to survive the Titanic sinking. It likely has to do with my age as the Pclass doesn't seem to have any affect.

Question 4 - Use your model to classify the 418 randomly selected passengers in the test set (titanic_test.csv) on github.

I believe this should be relatively straight forward..BWHAHAHAHAHAH!!! I'll load in the test set and cull the NAs and run it through the model.

```
```{r}
```

```
Titanic_418 <- read.csv(file.choose(), header = TRUE)
View(Titanic_418)
nrow(Titanic_418)
```

```
Titanic_418_new <- Titanic_418 %>% select(Pclass, Age)
Titanic_418_new
nrow(Titanic_418_new)
```

```
Titanic_418_final <- Titanic_418_new[complete.cases(Titanic_418_new),]
#summary(Titanic_418_final)
nrow(Titanic_418_final)
#nrow(train)
```

```
nrow(test_new$Survived)
```

```
knn.418_final <- knn(train, Titanic_418_final, train_new$Survived, k = 18)
knn.418_final
View(knn.418_final)
nrow(knn.418_final)
nrow(train_new)
summary(train_new$Survived)
train_new$Survived <- as.factor(train_new$Survived)
```

```
nrow(train_new$Survived)
nrow(knn.418_final)
```

```
#confusionMatrix(knn.418_final, train_new$Survived, k = 18)
```
```

****I'm stuck on this one. I'm going to create another .Rmd and attack Part 2****